Assignment 7

100 pts

There are two parts to this assignment. For the first part, you will carry out an SQL injection attack on a Chattr server. For the second, you will fix a number of security weaknesses in your own Chattr application that you developed for Assignment2. Your solution is due on November 30, 2017 no later than 10:00 P.M. PST. You may work with *one* other person in the class on this assignment, however, each student must submit his/her own solution. You may not discuss your solution with anyone except your partner until seven days after the assignment deadline. See Section 5 for additional information on submitting your solution. *You will be graded on both functionality and security requirements.*

1 Required Functionality

Your application must implement all the functionality of Assignment 2.

2 Part 1: SQL Injection

For this part of the assignment you will insert data into a database using an SQL injection attack. We are running a Chattr server at:

`http://gul027.ucsd.edu:4040/index.php`

Joey Pardella has created an account on this Chattr server. You can view hist posts at:

`http://gul027.ucsd.edu:4040/view.php?user=pardellaj`

Your goal is to insert the message "XXXXXXXX is l33t" on Joey's message board, where XXXXXXXX is your PGP key ID (8 hexadecimal digits). The post must obviously appear to be from Joey. For example, if the TA PGP key ID is `7864D1BD`,[1] then a successful post would look like this:

When	Who	What
2014-11-25 14:06:13.031256	pardellaj	7864D1BD is l33t

You may interact with the Chattr server in any way you wish. The assignment can be completed using SQL injection. You do not have access to the code or the schema, so it may take you several attempts to figure out how to do this. After you successfully insert the message, you might get an error message complaining that the user doesn't exist. So when you see this error, check back the post list of pardellaj to verify if you have already inserted your message.

3 Part 2: Securing Chattr

For this part of the assignment, you must fix any security vulnerabilities we have identified in your Chattr application from Assignment 2. Use the VM from the second assignment to develop your solution.

[1] The TA PGP key ID for this class is `10D37DBD`. Key ID `7864D1BD` was the key ID of the TAs who created this assignment in 2014.

3.1 The Report

You will receive a report on the vulnerabilities we found in your Assignment 2 submission. Use this report as a guide to determine which problems you need to fix. You will be graded on the same set of vulnerabilities, although the tests may be slightly different.

3.2 Password Storage

Your application must store passwords *salted*. This means that if you stored the password in the clear or stored the password hashed using a fixed hash function (e.g. SHA1), you will need to fix this part of the application.

3.3 Vulnerabilities

Your application must also protect against (a) SQL injection attacks, (b) cross-site scripting attacks, and (c) cross-site request forgery attacks. The report you will receive will indicate which vulnerabilities we found.

4 Grading

The first part of the assignment is worth 25 points; you will receive 25 points if you successfully post your PGP key ID to Joey's account, as described above.

The second part of the assignment is worth 75 points. We will use the same VM as in Assignment 2 to test your solution. Your solution *must pass all the functional tests* of Assignment 2. Therefore, you should first make sure you have a working solution for Assignment 2. If you are not sure if your solution passes the functional tests, your TAs will be happy to test your solution and let you know. We expect everyone to pass these tests.

If your solution passes the functional tests, your grade for this part of the assignment will depend on how many of the attacks we test succeed against your application. The more attacks succeed, the lower the grade. The attacks will be variations of SQL injection, cross-site scripting attacks, and cross-site request forgery, as identified in your report. Your assignment grade (out of 75 points) will be calculated as:

$$(\text{Part 1 points}) + (\text{Assignment 2 functionality points}) - (\text{Part 2 vulnerability points})$$

For example, if you complete Part 1 and turn in a Chattr application that does not work, you will receive $25 + 0 - 0 = 25$ points. If you don't complete Part 1 and turn in a Chattr application that passes all the functionality tests of Assignment 2 but fails all security tests, you will receive $0 + 75 - 75 = 0$ points. There will be a 15-point penalty if your solution is not correctly signed and encrypted as described below.

This programming assignment may be completed individually or with one other student in the class. If you work with another student, *both you and your partner* must complete Part 1 and submit a solution for Part 2. For the latter, you may start with either your own or your partner's Assignment 2solution. We will use the same VM we provided you, so make sure that your solution works on the original image of the VM. You may consult any online references you wish. If you use any code you find online, please document it in a `README` file submitted with your solution (see Section 5).

5 Submitting Your Solution

Your solution to the first part of this assignment is a successful post to Joey Pardella's Chattr page. Your attack must be completed by the assignment deadline (10:00 P.M. PST on November 30, 2017).

Your solution to the second part consists of the same PHP files you submitted for Assignment 2, namely `index.php`, `login.php`, `view.php`, `post.php`, and `logout.php`, as well as the database schema (`db.sql`).

You may submit additional PHP files if your solution relies on them. Your solution must by submitted via email to `cs127f1@ieng6.ucsd.edu` by November 30, 2017, 10:00 P.M. PST. It must be a `gzip`-compressed `tar` archive, signed with your PGP key and encrypted to the `cs127f1@ieng6.ucsd.edu` PGP key, which is provided on the CSE 127 Web page and has key fingerprint:

```
E1BF 1E04 1104 28DA 4F89 6543 B033 B3DC 10D3 7DBD.
```

You must send a plain email with the encrypted and signed archive file as an attachment. The email must have the subject "Homework 7 Submission" and the attachment must be named "`{PID}-hw7.tgz.asc`" (where `{PID}` is your PID). To create a `gzip`-compressed `tar` archive, copy the files you wish to submit to single directory, change into that directory, and issue the command:

```
tar -zcvf /path/to/archive/{PID}-hw7.tgz *.php *.sql
```

This will create an archive in the directory `/path/to/archive/` containing all the PHP and SQL files in the current working directory. To sign your submission with GPG:

```
gpg --encrypt --sign --armor -r cs127f1@ieng6.ucsd.edu {PID}-hw7.tgz
```

This will produce a file named "`{PID}-hw7.tgz.asc`" in the same directory. You will need to have imported the `cs127f1@ieng6.ucsd.edu` public key into your GPG keyring first.